SPECIFICATION AMENDMENTS

(Replace third full paragraph on page 2 with the following paragraph.) The use of cryptographic key pairs was disclosed in U.S. Pat. No. 4,200,770, entitled "CRYPTOGRAPHIC APPARATUS AND METHOD." U.S. Pat. No. 4,200,770 also disclosed the application of key pairs to the problem of key agreement over an insecure communication channel. The algorithms specified in this U.S. Pat. No. ~~4,200,700~~ 4,200,770 rely for their security on the difficulty of the mathematical problem of finding a discrete logarithm. U.S. Pat. No. 4,200,770 is hereby incorporated by reference into the specification of the present invention.

(Replace the first full paragraph on page 7 with the following paragraph.) It is an object of the present invention to efficiently create a digital signature using a modulus p selected from the following families of equations:

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $0<2c<=d$, where r ~~!=~~ does not equal to 1, and where $GCD(c,d)=1$, where GCD is a function that returns the greatest common denominator between the variables in parenthesis;

$$p=(2^{dk}-2^{(d-1)k}+2^{(d-2)k}-\ldots-2^{k}+1)/r,$$

where d is even, and where k is not equal to 2 (mod 4);

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $3d<6c<4d$, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{ck}+1)/r,$$

where $0<2c<=d$, where r ~~!=~~ does not equal to 1, and where $GCD(c,d)=1$; and

$$p=(2^{4k}-2^{3k}+2^{2k}+1)/r.$$

(Replace the first full paragraph on page 8 with the following paragraph.) The first step through sixth step are done by each user who wishes to have its message digitally signed. The first step is selecting a modulus p from the following family of equations:

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $0<2c<=d$, where r ~~!=~~ does not equal to 1, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{(d-1)k}+2^{(d-2)k}-\ldots-2^{k}+1)/r,$$

where d is even, and where k is not equal to 2 (mod 4);

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $3d<6c<4d$, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{ck}+1)/r,$$

where $0<2c<=d$, where r ~~!=~~ does not equal to 1, and where $GCD(c,d)=1$; and

$$p=(2^{4k}-2^{3k}+2^{2k}+1)/r.$$

(Replace the sixth full paragraph on page 9 with the following paragraph.) The seventh step through fifteenth step are done to identity the user who wishes to have a message digitally signed. The seventh step is a prover ~~retrieving~~ generating its private key ~~w~~ w_p and public key $W_p=w_pG$, and distributing its W_p.

(Replace the seventh full paragraph on page 9 with the following paragraph.) The eighth step is a verifier retrieving the prover's public key ~~W~~ W_p.

(Replace the eighth full paragraph on page 9 with the following paragraph.) The ninth step is the prover generating a private integer ~~k~~ k_p.

(Replace the ninth full paragraph on page 9 with the following paragraph.) The tenth step is the prover combining ~~*k*~~ $\underline{k}_p$ and ~~prover's~~ G to form K using the form of the ~~prover's~~ modulus p.

(Replace the twelfth full paragraph on page 9 with the following paragraph.) The thirteenth step is the prover combining c, ~~*k*~~ $\underline{k}_p$, and ~~*w*~~ $\underline{w}_p$ to form a response integer v.

(Replace the fourteenth full paragraph on page 9 with the following paragraph.) The fifteenth step is the verifier combining cG, K, and ~~*W*~~ $\underline{W}_p$ using the form of the ~~prover's~~ modulus p and checking to see if the combination is equal to vG. If the combination is equal to vG then the prover is properly identified. Otherwise, the prover is not properly identified.

(replace the fifteenth full paragraph on page 9 with the following paragraph.) The sixteenth step through the twenty-first step are done by the person digitally signing a message. The sixteenth step is a signer ~~retrieving~~ generating its private key ~~*w*~~ $\underline{w}_s$.

(Replace the sixteenth full paragraph on page 9 with the following paragraph.) The seventeenth step is the signer generating a private integer ~~*k*~~ $\underline{k}_s$.

(Replace the seventeenth full paragraph on page 9 with the following paragraph.) The eighteenth step is the signer combining k $\underline{k}_s$ and G to form K using the form of the ~~prover's~~ modulus p.

(Replace the second full paragraph on page 10 with the following paragraph.) The twentieth step is the signer combining h, k $\underline{k}_s$, and ~~w~~ $\underline{w}_s$ to form an integer s.

(Replace the fourth full paragraph on page 10 with the following paragraph.) The twenty-second step through the twenty-fifth step are done by the person verifying the digital signature. The twenty-second step is the verifier retrieving the prover's public key ~~W~~ $\underline{W}_p$.

(Replace the seventh full paragraph on page 10 with the following paragraph.) The twenty-fifth step is the verifier combining h, k $\underline{K}$, and ~~W~~ $\underline{W}_p$ using the form of the ~~prover's~~ modulus p and checking to see if the combination is equal to sG. If so, then the digital signature is verified. Otherwise, the digital signature is not verified.

(Replace the eighth full paragraph on page 10 with the following paragraph.) The twenty-sixth step through the thirty-first step are alternative steps for digitally signing a message. The twenty-sixth step is a signer retrieving its private key ~~w~~ $\underline{w}_s$.

(Replace the ninth full paragraph on page 10 with the following paragraph.) The twenty-seventh step is the signer generating a private integer ~~*k*~~ $\underline{k}_s$.

(Replace the tenth full paragraph on page 10 with the following paragraph.) The twenty-eighth step is the signer combining ~~*k*~~ $\underline{k}_s$ and G to form K using the form of the ~~prover's~~ modulus p.

(Replace the twelfth full paragraph on page 10 with the following paragraph.) The thirtieth step is the signer combining h, ~~*k*~~ $\underline{k}_s$, and ~~*w*~~ $\underline{w}_s$ to form an integer s.

(Replace the thirteenth full paragraph on page 10 with the following paragraph.) The ~~thirty-second~~ thirty-first step through thirty-sixth steps are alternative steps for verifying the digital signature of the alternative signing steps. The thirty-first step is the signer sending the message M and the digital signature (h,s) of M.

(Replace the fourteenth full paragraph on page 10 with the following paragraph.) The thirty-second step is the verifier retrieving the prover's public key ~~*W*~~ $\underline{W}_p$.

(Replace the first full paragraph on page 11 with the following paragraph.) The thirty-fourth step is the verifier combining h, ~~W~~ $\underline{W_p}$, and sG using the form of the ~~prover's~~ modulus p to form K.

(Replace the last paragraph on page 11 with the following paragraph.) The present invention is a method of identifying a user, generating a digital signature for a message of the user, and verifying the digital signature in an efficient manner (i.e., in fewer steps than the prior art) using a modulus p selected from the following family of equations:

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $0<2c<=d$, where r ~~!=~~ does not equal 1, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{(d-1)k}+2^{(d-2)k}-\ldots-2^{k}+1)/r,$$

where d is even, and where k is not equal to 2 (mod 4);

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $3d<6c<4d$, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{ck}+1)/r,$$

where $0<2c<=d$, where r != does not equal 1, and where $GCD(c,d)=1$; and

$$p=(2^{4k}-2^{3k}+2^{2k}+1)/r.$$

(Replace the last paragraph on page 13 with the following paragraph.) The first step 1 of the present method is selecting a modulus p from the following family of equations:

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $0<2c<=d$, where r != does not equal 1, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{(d-1)k}+2^{(d-2)k}-\ldots-2^{k}+1)/r,$$

where d is even, and where k is not equal to 2 (mod 4);

$$p=(2^{dk}-2^{ck}-1)/r,$$

where $3d<6c<4d$, and where $GCD(c,d)=1$;

$$p=(2^{dk}-2^{ck}+1)/r,$$

where $0<2c<=d$, where r != does not equal 1, and where $GCD(c,d)=1$; and

$$p=(2^{4k}-2^{3k}+2^{2k}+1)/r.$$

(Replace the seventh full paragraph on page 14 with the following paragraph.) The seventh step 7 of the method is a prover ~~retrieving~~ generating its private key ~~w~~ w_p and public key $W_p = w_p G$ and dirstibuting W_p.

(Replace the eighth full paragraph on page 14 with the following paragraph.) The eighth step 8 of the present method is a verifier retrieving the prover's public key ~~W~~ W_p.

(Replace the ninth full paragraph on page 14 with the following paragraph.) The ninth step 9 of the present method is the prover generating a private integer ~~k~~ k_p.

(Replace the first full paragraph on page 15 with the following paragraph.) The tenth step 10 of the present method is the prover combining ~~k~~ k_p and prover's G to form K using the form of the ~~prover's~~ modulus p.

(Replace the third full paragraph on page 15 with the following paragraph.) The twelfth step 12 of the present method is the verifier sending a challenge integer c to the prover.

(Replace the fourth full paragraph on page 15 with the following paragraph.) The thirteenth step 13 of the present method is the prover combining c, ~~k~~ $\underline{k}_p$, and ~~w~~ $\underline{w}_p$ to form a response integer v.

(Replace the sixth full paragraph on page 15 with the following paragraph.) The fifteenth step 15 of the present method is the verifier combining cG, K, and ~~W~~ $\underline{W}_p$ using the form of the ~~prover's~~ modulus p and checking to see if the combination is equal to vG. If the combination is equal to vG then the prover is properly identified. Otherwise, the prover is not properly identified.

(Replace the eighth full paragraph on page 15 with the following paragraph.) The sixteenth step 16 of the present method is a signer retrieving its private key ~~w~~ $\underline{w}_s$.

(Replace the ninth full paragraph on page 15 with the following paragraph.) The seventeenth step 17 of the present method is the signer generating a private integer ~~k~~ $\underline{k}_s$.

(Replace the tenth full paragraph on page 15 with the following paragraph.) The eighteenth step 18 of the present method is the signer combining ~~k~~ $\underline{k}_s$ and G to form K using the form of the ~~signer's~~ modulus p.

(Replace the twelfth full paragraph on page 15 with the following paragraph.) The twentieth step 20 of the present method is the signer combining h, ~~k~~ $\underline{k}_s$, and ~~w~~ $\underline{w}_s$ to form an integer s.

(Replace the second full paragraph on page 16 with the following paragraph.) The twenty-second step 22 of the present method is the verifier retrieving the prover's public key ~~W~~ $\underline{W}_p$.

(Replace the fifth full paragraph on page 16 with the following paragraph.) The twenty-fifth step 25 of the present method is the verifier combining h, ~~k~~ $\underline{K}$, and ~~W~~ $\underline{W}_p$ using the form of the ~~prover's~~ modulus p and checking to see if the combination is equal to sG. If so, then the digital signature is verified. Otherwise, the digital signature is not verified.

(Replace the seventh full paragraph on page 16 with the following paragraph.) The twenty-sixth step 26 of the present method is a signer retrieving its private key ~~w~~ $\underline{w}_s$.

(Replace the eighth full paragraph on page 16 with the following paragraph.) The twenty-seventh step 27 of the present method is the signer generating a private integer ~~k~~ $\underline{k}_s$.

(Replace the ninth full paragraph on page 16 with the following paragraph.) The twenty-eighth step 28 of the present method is the signer combining k $\underline{k}_s$ and G to form K using the form of the ~~signer's~~ modulus p.

(Replace the eleventh full paragraph on page 16 with the following paragraph.) The thirtieth step 30 of the present method is the signer combining h, k $\underline{k}_s$, and ~~w~~ $\underline{w}_s$ to form an integer s.

(Replace the second full paragraph on page 17 with the following paragraph.) The thirty-second step 32 of the present method is the verifier retrieving the prover's public key ~~W~~ $\underline{W}_p$.

(Replace the fourth full paragraph on page 17 with the following paragraph.) The thirty-fourth step 34 of the present method is the verifier combining h, ~~W~~ $\underline{W}_p$, and sG using the form of the ~~prover's~~ modulus p to form K.

(Replace the first full paragraph on page 24 with the following paragraph.) A method of identifying user, generating digital signature, and verifying digital signature by selecting a modulus p in the form of $p=(2^{dk}-2^{rk}-1)/r$; $p=(2^{dk}-2^{(d-1)k}+2^{(d-2)k}-\ldots-2^{k}+1)/r$; $p=(2^{dk}-2^{rk}-1)/r$; $p=(2^{dk}-2^{rk}+1)/r$; and $p=(2^{dk}-2^{3k}+2^{2k}+1)/r$; selecting an elliptic curve E and an order q; selecting a basepoint G; generating a private key w; generating a public key $W=wG$; distributing p, E, q, G,

and W to at least a prover, a verifier, and a signer; ~~retrieving~~ generating ~~a~~ the prover's private key ~~w~~ w_p and public key $W_p = w_p G$; retrieving the prover's public key ~~W~~ W_p; generating a private integer ~~k~~ k_p; combining ~~k~~ k_p and ~~the prover's~~ G to form K using ~~the prover's modulus~~ p; sending K to the verifier; sending a challenge integer c to the prover; combining c, ~~k~~ k_p, and ~~w~~ w_p to form a response integer v; sending v to the verifier; combining cG, K, and ~~W~~ W_p using ~~the prover's modulus~~ p and checking to see if the combination is equal to vG. If not so, stop. Otherwise, ~~retrieving~~ generating, by the signer, the signer's private key ~~w~~ w_s; generating a private integer ~~k~~ k_s; combining ~~k~~ k_s and G to form K using ~~the prover's modulus~~ p; combining K and a message M to form an integer h; combining h, ~~k~~ k_s, and ~~w~~ w_s to form an integer s; sending M and (K,s) as a digital signature of M; retrieving ~~the prover's public key~~ ~~W~~ W_p; receiving M and (K,s); combining K and M to form an integer h; and combining h, ~~k~~ K, and ~~W~~ W_p using ~~the prover's modulus~~ p and checking to see if the combination is equal to sG. If so, the digital signature is verified.